



## 【特許請求の範囲】

【請求項1】 複数の装置が、個別の装置間で他の装置には秘密に情報の通信を行なうための秘密通信路と、各装置から他の全ての装置へ共通に情報を送信するための放送通信路とを介して接続された通信システムにおいて、

署名者グループ内の各装置が、第1の秘密情報をランダムに選び、前記グループ内の各装置に秘密に分散させ、各装置が、前記第1の秘密情報に所定の第1の関数を作

用させ、得られた出力値を前記グループ内の全装置に放送し、  
各装置の前記第1の秘密情報を分散加算し、  
各装置の前記出力値を分散乗算し、当該乗算の結果とメッセージとを所定の第2の関数で処理し、  
該処理結果と、前記分散加算の結果と、公開された元とを用いて第2の秘密情報を分散演算し、  
分散された第2の秘密情報を復元し、前記分散乗算結果と共に署名として出力することを特徴とする署名生成方法。

【請求項2】 前記第2の秘密情報の分散演算が線形結合であることを特徴とする請求項1に記載の署名生成方法。

【請求項3】 前記第2の秘密情報の分散演算がべき乗を含む積による結合であることを特徴とする請求項1に記載の署名生成方法。

【請求項4】 前記第2の秘密情報の分散演算が線形結合と乗算との組み合わせであることを特徴とする請求項1に記載の署名生成方法。

【請求項5】 前記第2の関数が一方向性の関数であることを特徴とする請求項1に記載の署名生成方法。

【請求項6】 前記第1の秘密情報は所定の有限体上の元であり、各演算は該有限体上の演算を利用することを特徴とする請求項1に記載の署名生成方法。

【請求項7】 前記第1の秘密情報の分散を、  
該第1の秘密情報を有する第1の装置が、当該秘密情報から所定の部分配列を生成し、  
前記第1の装置が、前記部分配列より他の装置の各々に対する第1の部分情報を夫々抽出して、各装置に前記秘密通信路を介して送信し、  
前記第1の装置が、前記第1の部分情報に所定の関数を作

用させ、得られた出力値を、各装置に前記放送通信路を介して放送し、  
前記各装置が、前記第3の部分情報と放送された第2の部分情報とを比較して、前記第1の装置による秘密の分散を確認することを特徴とする請求項1に記載の署名生成方法。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、通信路によって接続された複数の加入者の内の何人かの加入者によるグループにおけるデジタル署名を、そのグループに参加する加入者間で分散生成する方式に関するものである。

## 【0002】

【従来の技術】 通信路によって接続された複数の情報処理装置を含む情報通信システムにおいて、送信された情報が指定された受信装置以外に漏れないこと（情報の秘匿）を保証するために役立つ技術の一つとして、暗号技術が知られている（池野、小山：“現代暗号理論”、p. 224-225、電子情報通信学会）。

【0003】 暗号技術は、上述した情報の秘匿機能の他に、その通信システムにおいて受け取った情報が、指示された装置から発信されたこと（途中で改竄されなかったこと）を確認できる認証機能、及びその受け取った情報が指示された装置から発信されたことを第三者にも証明できるデジタル署名と呼ばれる機能を実現するために有効であることもよく知られている。

【0004】 特に、公開鍵暗号方式の1つであるRSA暗号を用いた認証及びデジタル署名方式は広く用いられている（例えば、R. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, 21, 2, 1978, pp. 120-126. または米国特許4,405,828号参照）。

【0005】 また、RSA暗号以外の公開鍵暗号方式によるデジタル署名方式の1つとして、A. Fiat, A. Shamirによって提案された方式がよく知られている（"How to Prove Yourself: Practical Solutions to Identification and Signature Problems", Advances in Cryptology —Crypto'87, Lecture Notes in Computer Science, 263, Springer-Verlag, 1988, pp. 186-194. または米国特許4,748,668号参照）。この方式では、安全な識別及び与えられたメッセージのデジタル署名を実現するために次の処理を行う。

【0006】 (1) 与えられたメッセージのデジタル署名を計算する装置、あるいは通信システムにおいて信頼できるセンタの役割を果たす装置が、 $1, \dots, N-1$ （但し  $N$  は2つの素数  $p, q$  の積である）の中から元  $a$  をランダムに選び、署名を計算する装置の秘密情報とする。

【0007】 (2) その秘密  $a$  を選んだ装置は、 $a^l \bmod N$ （但し、 $l$  は  $\gcd(l, \lambda(N))=1$  を満たし、ここで、 $\lambda(N)=\text{lcm}(p-1, q-1)$ 、 $\gcd(a, b)$  は  $a, b$  の最大公約数、 $\text{lcm}(a, b)$  は  $a, b$  の最小公倍数を意味する）を計算し、そ

の装置によって生成されたデジタル署名を確認するための公開情報とする。

【0008】(3) その装置によって与えられたメッセージ $m$ のデジタル署名の生成処理において、 $\{1, \dots, N-1\}$ の内からランダムに選ばれた秘密の元 $r$ を用いて計算された $R=r^{-1} \bmod N$ と、与えられた公開のメッセージ $m$ を連続させた値 $R|m$ を入力として所定の関数 $h$ を用いて $e=h(R|m)$ を計算する。それらを入力として $s=r*a^e \bmod N$ の計算を実行することによって得られた出力 $s$ 及び $R$ を、与えられたメッセージのデジタル署名とする。

【0009】(4) ある与えられたメッセージ $m$ に対するデジタル署名 $(s, R)$ を確認するためには、 $s^{-1} \bmod N$ と $R*(a^{-1})^{(h(R|m))} \bmod N$ の計算を実行し、それらの結果が等しいことを確認する。

【0010】また、G.P. Schnorr によって提案された安全な識別及び与えられたメッセージのデジタル署名を実現する方式("Efficient Identification and Signatures for Smart Cards", Advances in Cryptology—Crypto'89, Lecture Notes in Computer Science, 435, Springer-Verlag, 1990, pp. 239-252. または米国特許4,995,082号参照)は、次の処理を行う。

【0011】(1) 与えられたメッセージのデジタル署名を計算する装置、あるいは通信システムにおいて信頼できるセンタの役割を果たす装置は $\{1, \dots, p\}$ (但し $p$ は素数である)の内から元 $a$ をランダムに選び、署名を計算する装置の秘密情報とする。

【0012】(2) その秘密 $a$ を選んだ装置は、 $g^{-a} \bmod q$ (但し、 $q$ は $p$ が $q-1$ の除数になるような素数であり、有限体 $GF(q)$ に属する元 $g$ の位数が $p$ となる)を計算し、その装置によって生成されたデジタル署名を確認するための公開情報とする。

【0013】(3) その装置によって与えられたメッセージ $m$ のデジタル署名の生成処理において、 $\{1, \dots, p\}$ の内からランダムに選ばれた秘密の元 $r$ を用いて計算された $R=g^r \bmod q$ と、与えられた公開のメッセージ $m$ を連続させた値 $R|m$ を入力として所定の関数 $h$ を用いて $e=h(R|m)$ を計算する。それらを入力として $s=r+a*e \bmod p$ の計算を実行することによって得られた出力 $s$ 及び $R$ を、与えられたメッセージのデジタル署名とする。

【0014】(4) ある与えられたメッセージ $m$ に対するデジタル署名 $(s, R)$ を確認するためには、 $h(g^{-s}((g^{-(-a)})^{(h(R|m))}) \bmod q) | m) R | m)$ の計算(但し $x^y$ は $x$ の $y$ 乗を表す)を実行し、この結果と $e$ とが等しいことを確認する。

【0015】また、T. ElGamal によって提案された安全な識別及び与えられたメッセージのデジタル署名を実現する方式("A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Tra-

nsactions on Information Theory, IT-31, 4, 1985, p. 469-472. American National Standard X9.30-199x, Digital Signature Algorithm, Feb. 1992参照)は、次の処理を行う。

【0016】(1) 与えられたメッセージのデジタル署名を計算する装置、あるいは通信システムにおいて信頼できるセンタの役割を果たす装置、は $\{1, \dots, p\}$ (但し $p$ は素数である)の内から元 $a$ をランダムに選び、署名を計算する装置の秘密情報とする。

【0017】(2) その秘密 $a$ を選んだ装置は、 $g^{(-a)} \bmod q$ (但し、 $q$ は $p$ が $q-1$ の除数になるような素数であり、 $GF(q)$ に属する元 $g$ の位数が $p$ となる)を計算し、その装置によって生成されたデジタル署名を確認するための公開情報とする。

【0018】(3) その装置によって与えられたメッセージ $m$ のデジタル署名の生成処理において、 $\{1, \dots, p\}$ の内からランダムに選ばれた秘密の元 $r$ を用いて計算された $R=g^r \bmod q$ と、所定の関数 $h$ を用いて与えられた公開のメッセージ $m$ を入力として得られた値 $e=h(m)$ とを入力として、 $s=(e+R*a)*r^{-1} \bmod p$ の計算を実行することによって得られた出力 $s$ 、及び $R$ を与えられたメッセージのデジタル署名とする。

【0019】(4) ある与えられたメッセージ $m$ に対するデジタル署名 $(s, R)$ を確認するためには、 $(g^{(-a)})^R (g^r)^s$ と $g^m \bmod q$ の計算を実行し、それらの結果が等しいことを確認する。

【0020】一方、前述のように守秘及び認証を実現する情報通信システムにおいて、秘密情報を守りながら信頼性を高める手段として、秘密情報を分散し、与えられたメッセージのデジタル署名の計算を通信路によって接続された複数の計算装置(以下、署名者グループといい、そのグループに参加する各計算装置を加入者という、またそのグループに参加する加入者の数を $n$ で表す)の間で分散する方法がY. Desmedt, Y. Frankel によって提案されている("Threshold Cryptosystems", Advances in Cryptology—Crypto'89, 435, Springer-Verlag, 1990, pp. 307-315; "Shared Generation of Authenticators and Signatures", Advances in Cryptology—Crypto'91, 576, Springer-Verlag, 1992, pp. 457-469参照)。

【0021】この分散デジタル署名方式の基本的な部分は、複数の加入者からなる通信システムにおいて秘密情報を分散する方式である。具体的に、ある秘密情報 $x$ が複数の加入者間で分散保持されるとは、以下の条件(a), (b)が満たされるように各加入者 $i$ が秘密情報 $x$ に対応する部分情報 $x_i$ を生成し、他の加入者に分配することを意味する。

【0022】(a) 秘密情報 $x$ を復元するために $t+1$ 人の加入者の部分情報が必要である。以後、その秘密情報を復元するための必要な加入者の数 $t+1$ をしきい値と呼

ぶ。

【0023】(b) しきい値未満の数 ( $t$  以下) の部分情報ではその秘密情報に関するどのような情報も得ることができない。

【0024】従来の基本的な秘密分散方式は、A. Shamir によって提案され "How to Share a Secret", Communications of the ACM, Vol. 22, 11, 1979 参照)、次のように実現された。すなわち、ある加入者の情報を秘密に複数の加入者に分散するために、定数項が前述の秘密情報となる  $t$  次の多項式  $f(x)$  をランダムに選び、 $n$  個の異なる値に対するその多項式の値  $f(i)$  ( $i=1, \dots, n$ ) を各加入者に配る。この加入者  $i$  に配られる多項式の値  $f(i)$  が前述の部分情報の一部分になる。よって、秘密情報は  $t+1$  個の部分情報を用いた多項式補間によって復元できる ( $t$  以下の部分情報では秘密情報に関するどのような情報も得ることはできない)。

【0025】前述の Y. Desmedt, Y. Frankel による秘密分散方式に基づく RSA 暗号を用いた分散型デジタル署名方式は以下の条件 (i), (ii) を満たす。

【0026】(i) 署名者グループに対する与えられたメッセージのデジタル署名を生成するために  $t+1$  人の加入者の協力があれば十分である。

【0027】(ii) しきい値未満の数 ( $t$  以下) の加入者では与えられたメッセージのデジタル署名を生成できない。

【0028】但し、(i), (ii) の条件だけではデジタル署名生成処理が分散されたとき、 $t+1$  以上の加入者が協力しても不正な加入者があった場合には、署名を生成できないことがある。

【0029】それに対して、どのような誤りを持つ加入者に対してもデジタル署名を生成できるような方式は、確認可能な秘密分散方式に基づいて構成できることが知られている。どのような誤りを持つ加入者にも耐えられる秘密分散方式として提案された確認可能な秘密分散 (Verifiable Secret Sharing) とは、前述の条件 (a), (b) に次の条件 (c), (d) を付加することによって定義される。

【0030】(c) 不正な部分情報と正しい部分情報とが混在していても、 $t+1$  個の正しい部分情報があれば元の秘密情報を復元するために十分である。

【0031】(d) 全ての加入者がその秘密の部分情報を受け取ったとき、その部分情報がある秘密情報  $x$  を復元するために正しい情報であるかどうかを確認できる。

【0032】前述の条件 (c), (d) を満たす確認可能な秘密分散方式に基づいて、次の条件を満たす分散デジタル署名方式を構成できる。

【0033】(i') 不正な加入者が正しい加入者と混在していても与えられたメッセージのデジタル署名を生成するために  $t+1$  人の正しい加入者の協力があれば十分である。

【0034】(ii) しきい値未満の数 ( $t$  以下) の加入者では与えられたメッセージのデジタル署名を生成できない。

【0035】秘密の通信路を持つ通信システムに対して、全加入者の  $3$  分の  $1$  より少ない数であればどのような誤りを持つ加入者にも耐えられる確認可能な秘密分散方式 (しきい値  $t$  が  $t < n/3$  を満たす場合) を構成するためには、従来の誤り訂正符号技術で十分であることが、M. Ben-Or, S. Goldwasser, A. Wigderson によって示されている ("Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation", ACM STOC 1988 参照)。

【0036】更に、全加入者の半分以上より少ない数であれば、どのような誤りを持つ加入者にも耐えられる確認可能な秘密分散方式を構成するためには、条件を更に追加する必要があり、全ての加入者が同じメッセージを受信したことを確認できる放送型通信路を、全ての加入者が持つとした場合、次の2通りの構成方法が知られている。

【0037】(1) 零知識対話証明システム (辻井、笠原: "暗号と情報セキュリティ", 昭晃堂、1990年、参照) で用いられている 'Cut and Choose' と呼ばれる技術を利用し、前述の A. Shamir による基本的な秘密分散方式によって元の秘密  $s$  を分散した上に、配られた部分情報  $s_i$  ( $i=1, \dots, n$ ) を更に分散する方式。つまり、確認可能な分散方式によって配られた全ての部分情報は、秘密部分  $s_i$  に対する秘密部分になるように生成された部分行列と考えることができる。

【0038】但し、前述の 'Cut and Choose' 技術を用いることで、前述の条件 (d) による確認とは統計的確認になり、秘密情報を復元できる正しい分散が行われているかどうかを示す判定出力に誤り確率が生じる。但し、その誤り確率は設定する安全性のパラメータにより無視できる程度まで小さくすることができる。具体例として、T. Rabin, M. Ben-Or による方式 ("Verifiable Secret Sharing and Multiparty Protocols with Honest Majority", ACM STOC 1989 参照) があげられる。

【0039】(2) 非対話型で特殊な代数的な性質を持つ一方向性関数を利用する方法。但し、このように構成された秘密分散方式の安全性は、その代数的な性質を満たす一方向性関数の逆元を計算するのが困難である (実用的な逆元計算方式が存在しない) という暗号的な仮定をする必要が生じる。具体例は、P. Feldman によって提案された ("A Practical Scheme for Non-Interactive Verifiable Secret Sharing", IEEE FOCS, 1987 参照)。

【0040】また、上述の確認可能な秘密分散方式を利用することによって、デジタル署名の分散生成処理を含めた、ある与えられた有限体上の分散演算を安全に実現する回路を構成できることが T. Rabin, M. Ben-Or

(“Verifiable Secret Sharing and Multiparty Protocols with Honest Majority”, ACM STOC, 1989 参照)、D. Beaver (“Secure Multiparty Protocols and Zero-Knowledge Proof Systems Tolerating a Faulty Minority”, Journal of Cryptology, 1991, 4, pp. 75-122: “Efficient Multiparty Protocols Using Circuit Randomization”, Advances in Cryptology-Crypto'91, 1992 参照) 及び M. Franklin, S. Haber (“Joint Encryption and Message-Efficient Secure Computation”, Advances in Cryptology-Crypto'93, 1994 参照) によって示されている。

【0041】このような確認可能な秘密分散方式を用いた分散演算回路によって、前述のような種々のデジタル署名方式に対する分散生成処理システムも構成できることが知られている。

【0042】

【発明が解決しようとしている課題】 上述の条件(i), (ii)を満たす分散デジタル署名方式に必要な通信量及び計算量は実用的であることが知られている。しかし、前述したように署名の分散生成処理に参加する加入者が不正を行った場合に署名の生成ができないことが有り得る。

【0043】それに対して、上述の確認可能な秘密分散方式及び分散演算回路を利用する条件(i'), (ii')を満たす分散デジタル署名方式は、以下のように非実用的な通信量または計算量が必要であることが知られている。例えば、対話型確認可能な秘密分散方式(i)において1ビットを分散するために必要な通信量は、安全性のパラメータを $k$  (通常は100程度の値が用いられる)と表したとき $n$ 個の秘密部分に対して $n^2 k^2$ のオーダーとなることが知られており、効率的ではない。

【0044】また、一方向性関数を用い、通信量の少ない非対話型確認可能な秘密分散方式(2)は、 $n$ 個の秘密部分に対して $n$ 回のオーダーの特殊な一方向性関数の計算処理を行う必要があり、特に秘密分散処理を安全に分散計算を実行するための部分処理として用いる場合には、実行すべき秘密分散処理の数が多くなり(例えば、分散乗算においては $n^2$ のオーダーとなる)、全体的に非実用的な計算量になる。

【0045】以上のように、上述の従来技術による対話型確認可能な秘密分散方式(i)に基づいた分散デジタル署名は通信量が非常に大きく、非対話型確認可能な秘密分散方式(2)に基づいた分散デジタル署名は計算量が非常に大きいという問題があった。

【0046】本発明は、前述の対話的な方式(i)及び非対話的な方式(2)の中間に位置し、必要な計算量と通信量の両方が実用的なオーダーになる確認可能な秘密分散方式を利用し、条件(i)、(不正を行う加入者があれば署名を生成できないことが有り得る方式)及び条件(i') (あるしきい値以下の不正を行う加入者の数があっても

署名を生成できる方式)の中間に位置する分散デジタル署名方式(あるしきい値以下の不正を行う加入者の数があった場合、署名を生成できないことが有り得るが、不正を行った加入者は識別できる方式)を提案する。

【0047】

【課題を解決するための手段】 上述の課題を解決するために、本発明の署名生成方法は、複数の装置が、個別の装置間で他の装置には秘密に情報の通信を行なうための秘密通信路と、各装置から他の全ての装置へ共通に情報を送信するための放送通信路とを介して接続された通信システムにおいて、署名者グループ内の各装置が、第1の秘密情報をランダムに選び、前記グループ内の各装置に秘密に分散させ、各装置が、前記第1の秘密情報に所定の第1の関数を用いて作用させ、得られた出力値を前記グループ内の全装置に放送し、各装置の前記第1の秘密情報を分散加算し、各装置の前記出力値を分散乗算し、当該乗算の結果とメッセージとを所定の第2の関数で処理し、該処理結果と、前記分散加算の結果と、公開された元とを用いて第2の秘密情報を分散演算し、分散された第2の秘密情報を復元し、前記分散乗算結果と共に署名として出力することとを特徴とする。

【0048】

【実施例】 以下、図面を参照して本発明の実施例を詳細に説明する。

【0049】本実施例では、各加入者が、他の個々の加入者にメッセージを秘密に送ることができる秘密通信路と、全ての加入者が同じメッセージを受けたことを確認できる放送型通信路とによって接続される分散システムにおいて、図3に示すように、後述する秘密部分行列を生成し、その秘密の部分情報を分配すると同時に、後述する部分情報の認証子として用いられる一方向性ハッシュ値(但し、特殊な代数的な性質を持つ必要はない)を放送し、後述するCut and Choose処理を行う秘密分散方式を用いて、署名者グループに参加する各加入者によってランダムに選ばれた秘密の元をそのグループの全ての加入者間で分散し、正しく分散されたランダムな秘密の分散演算処理を実行することとを特徴する。

【0050】まず、本実施例で利用する技術及び用語のいくつかについて説明する。

【0051】図4は、秘密 $s$ と部分行列 $S$ との関係を示す図である。

【0052】本実施例において、ある秘密 $s$ に対する部分行列 $S$ とは、従来の対話型確認可能な秘密分散方式に用いられている部分行列に、以下に説明するように、更に条件が付けられた秘密部分の $n \times n$ 行列 $S = [s(i, j)]$  ( $i, j = 1, \dots, n$ ) である。

【0053】すなわち、各行ベクトル  $(S_r(i) = [s(i, 1), \dots, s(i, n)])$ ,  $i = 1, \dots, n$  が、ある秘密 $s_r(i)$ に対する秘密部分ベクトル(ベクトルの要素 $j$ が、ある秘密分散方式における加入者 $j$ の秘密部分である)になり、

各列ベクトル  $(S_c(j) = [s(1, j), \dots, s(n, j)], j=1, \dots, n)$  が、ある秘密  $s_c(j)$  に対する秘密部分ベクトルになる。但し、前述の秘密のベクトル  $[s_r(1), \dots, s_r(n)]$  及び  $[s_c(1), \dots, s_c(n)]$  の両方は、元の秘密  $s$  に対する秘密部分ベクトルになる。

【0054】秘密分散処理のときに、このように構成された部分行列の列ベクトル  $i$  ( $i=1, \dots, n$ ) 及び秘密  $s_r(i)$  を、加入者  $i$  の秘密部分情報として各加入者  $i$  に送信することによって、秘密復元処理のときに、部分情報を配った加入者が不正をしなれば、各加入者によって放送された部分情報が正しいかどうかを非常に高い確率で確認できる。

【0055】また、本実施例では、元の秘密情報を持ち、秘密部分情報を配った加入者が不正をした場合でも、秘密復元処理のとき放送される部分情報が正しいかどうかを確認するために、秘密部分情報を配る上で各加入者  $i$  ( $i=1, \dots, n$ ) に対応する秘密  $s_r(i)$  の認証子のような役割を果たす値を、一方向性ハッシュ関数（池野、小山：“現代暗号理論”、pp. 224-225、電子情報通信学会、1986、参照）によって生成する。

【0056】次に、この一方向性ハッシュ関数について説明する。

【0057】一方向性ハッシュ関数とは、データ圧縮型スクランブルを行う関数であり、入力値から出力値を求める演算は容易であるが、出力値から入力値を求める逆演算は困難である関数である。但し、本実施例では、従来の非対話型確認可能な秘密分散方式に用いられる一方向性関数に比べると、特殊な代数学的な性質を持たなくてもよく、計算処理の高速な一方向性ハッシュ関数を利用できる。

【0058】一方向性ハッシュ関数の具体例として、R. Merkle によって DES (Data Encryption Standard) のようなブロック暗号を用いた一方向性ハッシュ関数が提案されている (“One Way Hash Functions and DES”, Advances in Cryptology - Crypto'89, Lecture Notes in Computer Science, Vol. 435, Springer-Verlag, 1990 参照)。

【0059】図6は、一方向性ハッシュ関数の具体的な構成を説明する図である。

【0060】同図において、(a) は、DES によるブロック暗号化を示しており、61は、64ビットの入力及び56ビットの鍵から64ビットの出力が得られる暗号化回路（図8では、DES を  $E$  で表している）である。

【0061】(b) は、このDES を部分処理として利用し、入力の長さが119ビットで出力の長さが112ビットとなる関数  $F$  の処理を示しており、62は、関数演算回路である。この処理は、次のように定義される。

【0062】まず、入力を二つの部分  $k, x$  に分ける（ただし、その一つの部分  $k$  の長さを55ビットとし、残りの部分  $x$  の長さを64ビットとする）。次に、その部分  $x$  を  $DE$

$S$  の入力とし、残りの部分  $k$  と '0' とを連続した値 '0',  $k$  を56ビットの鍵として得られた出力と、 $x$  とのXOR を計算した結果を、関数  $F$  の出力における左側の64ビットとする。同様に、同じ部分  $x$  を入力とし、残りの部分  $k$  と '1' とを連続した値 '1',  $k$  を鍵として得られた出力と、 $x$  とのXOR を計算した結果（64ビット）より48ビットを関数  $F$  の残りの部分（右側の48ビット）とする。これによって得られた二つの出力を連続した結果が112ビットの関数  $F$  の出力となる。

【0063】(c) は、与えられたメッセージを入力として、一方向性ハッシュ関数のハッシュ値を出力とする処理を示しており、63は、ハッシュ関数処理部である。この処理は、次のように実行される。

【0064】まず、与えられたメッセージの最初の119ビットを上記の関数  $F$  の入力として、最初の112ビットの出力を得る。次に、その出力を再び入力の112ビット分とし、以後メッセージから残りの7ビット分を連続して繰り返し関数  $F$  に入力する。最後に、全メッセージを入力したときに（メッセージの最後の7ビットを入力するために足りないビットがあれば適当に '0' を加える）得られた112ビットの出力を、メッセージに対するハッシュ値とする。

【0065】前述のように計算されたハッシュ関数の一方向性（1つのハッシュ値が与えられたとき、同じハッシュ値が得られるような異なる入力のメッセージを求めることが極めて困難であること）は、用いられるDES のようなブロック暗号の安全性（入力が与えられたとき出力は鍵によるランダムな変数になり、出力が与えられたとき入力は鍵によるランダムな変数になる）によって与えられることが、R. Merkle より示されている（前述の論文参照）。

【0066】更に、同じ論文では前述のハッシュ関数より効率のよい一方向性ハッシュ関数も提案されている。また、ブロック暗号を利用しない効率的な一方向性ハッシュ関数がR. Rivestによって提案されている (“The MD4 message digest algorithm”, Advances in Cryptology - Crypto'90, Lecture Notes in Computer Science, Vol. 537, Springer-Verlag, 1991. NIST Federal Information Processing Standard for Secure Hash, American National Standard X9.30-199x参照)。

【0067】次に、本実施例におけるCut and Choose処理を説明する。

【0068】従来の対話型確認可能な秘密分散方式で行われるCut and Chooseのように、秘密分散処理のとき全ての加入者が受け取った秘密部分情報が、確認可能に分散された秘密の部分情報となるかどうかを確認するために、元の秘密  $s$  に対する部分行列及びハッシュ値を分配すると同時に  $k$  個のランダムに選ばれた秘密  $l_1, \dots, l_k$  に対する部分行列及びハッシュ値を分配し、全加入者によってランダムに決められた  $k/2$  の秘密 ( $l_i(1), \dots, l_i$

( $k/2$ ) )に関する全ての秘密情報を放送し、残りの $k/2$ の秘密 ( $l_j(1), \dots, l_j(k/2)$ ) に対して  $l_j(1)+s, \dots, l_j(k/2)+s$  に関する全ての秘密情報を放送し、全ての放送された情報の中には誤りを持つ部分の数が  $t$  より多ければ、秘密分散処理が正しくないと判断される。

【0069】以上によって、従来の確認可能な秘密分散方式の条件(c), (d)の代わりに、次の条件(c'), (d')を満たす秘密分散方式が実現でき、それを用いて安全に分散計算を実行する方式及び通信システムが実現できる。

【0070】(c')不正な部分情報が正しい部分情報と混在したとき、 $t+1$ 個の正しい部分情報があっても元の秘密情報を復元できない場合が存在するが、不正をした加入者の識別は可能である。

【0071】(d')全ての加入者がその秘密の部分情報を受け取ったとき、その部分情報がある秘密情報 $x$ を復元するための正しい情報でなければ、復元処理のときに不正をした加入者の識別が可能である。

【0072】これらの条件は、従来方式のように不正な加入者があっても秘密を復元、すなわち誤り訂正は行えないが、不正な加入者は識別、すなわち誤り検出は可能にするものである。条件(c), (d)があれば不正な加入者がいても正しく分散演算が実行できる(正当性に対するフォールトトレランスを実現できる)が、条件(c'), (d')に変更しても不正な加入者は検出できるので警告などを行い再実行することによって正しい出力を得ることができ、正当性に対するフォールトトレランスを実現することができる。

【0073】よって、本実施例では、分散された秘密を復元するとき不正な加入者があった場合には、復元できないこともありえるが、不正を行った加入者の識別は可能であるような確認可能な秘密分散方式を実現できる。

【0074】そして、この秘密分散方式を利用して、署名者グループに参加する各加入者がランダムに選ばれた秘密の元をそのグループに参加する全加入者間で分散し、分散された秘密を入力として分散演算の実行から得られた分散出力の復元処理によって署名を生成する。

【0075】従って、本実施例では、不正を行った加入者の識別が可能であり、計算量と通信量の両方が実用的なオーダーになる分散デジタル署名方式が実現できる。

【0076】図1は、本発明の1実施例である、分散した情報処理装置を有する情報処理システムを示す図である。

【0077】同図において、11は、システムの各加入者の利用する情報処理装置である。以下の説明では、各装置とそれを利用する各加入者とを同一視して、加入者と呼ぶことにする。12は、全ての加入者に情報を公開することができる放送型通信路であり、13は、各加入者毎に秘密の通信を行うことができる秘密通信路である。

【0078】図2は、情報処理装置11のブロック構成を示す図である。同図において、21は、放送型通信路12または秘密通信路13により他の装置と通信を行なうための通信部である。22は、演算処理部であり、記憶部24のプログラムに従って、上述した一方向性ハッシュ関数などの各種演算や判定の処理を行なうとともに、装置各部を制御する。23は、例えば疑似乱数発生器のような乱数発生部であり、ランダムな値を生成するために利用される。24は、演算処理部22が実行すべきプログラムや、処理の過程で生成される演算結果等の情報や、他の装置から受信した情報、各種パラメータなどを記憶するための記憶部である。

【0079】以下、上述した装置構成により、分散デジタル署名を実現する方法を具体的に説明する。

【0080】〔実施例1〕この実施例では前述の C. P. Schnorr によって提案されたデジタル署名方式を用いた分散デジタル署名方式を実現する具体的な構成について述べる。

【0081】まず、 $\{1, \dots, p-1\}$ の内にある秘密の元を確認可能に分散する具体的な構成を次に示す。

【0082】ある与えられた秘密 $s$ に対する部分行列の具体的な説明を行う(図4参照)。 $\{1, \dots, p-1\}$ の内にある秘密の元 $s$ に対する部分行列 $S = [s(i, j)]$ ,  $i, j = 1, \dots, n$ とは、各行ベクトル  $(S_r(i) = [s(i, 1), \dots, s(i, n)]$ ,  $i = 1, \dots, n$ )の要素が $s_r(i)$ を定数項とする $t$ 次の多項式 $f_i$ の $n$ 個の異なる値 $i_1, \dots, i_n$ に対する値 $f_i(i_1), \dots, f_i(i_n)$ になり、各列ベクトル  $(S_c(j) = [s(1, j), \dots, s(n, j)]$ ,  $j = 1, \dots, n$ )の要素が $s_c(j)$ を定数項とする $t$ 次の多項式 $g_j$ の $n$ 個の異なる値 $j_1, \dots, j_n$ に対する値 $g_j(j_1), \dots, g_j(j_n)$ になり、さらに前述の値のベクトル  $[s_r(1), \dots, s_r(n)]$  及び  $[s_c(1), \dots, s_c(n)]$ の両方は、元の秘密 $s$ を定数項とする $t$ 次の多項式 $f$ 及び $g$ の値 $f(i_1, \dots, i_n)$ 及び $g(j_1, \dots, j_n)$ になっているものである。

【0083】秘密の元 $s$ を署名者グループに参加する全加入者間で秘密に分散保持されるように秘密部分を分配する秘密分散処理と、そのように分散された秘密または(不正があった場合に)不正をした加入者を明らかにする秘密復元処理の二つの処理に分けて説明する。

【0084】以下に、 $h$ は効率的な(高速な計算方式のある)一方向性ハッシュ関数を表す。例えば高速なブロック暗号化関数によって構成されたハッシュ関数(前述の"現代暗号理論"参照)が用いられる。安全性パラメータ $k$ はある定数 $k'$ に対して $k = nk'$ を満たす。この場合、秘密分散処理の確認が誤る確率は、Cut and Choose 処理によって(前記 T. Rabin, M. Ben-Or "Verifiable Secret Sharing and Multiparty Protocols with Honest Majority"参照)  $2^{-(k'-(t+1))}$ になる。

【0085】(1)秘密分散処理(図5参照):秘密の元 $s$ を持っている加入者 $d$ が $s$ に対する秘密部分を分配す

る処理

(ラウンド1) 加入者d は(図5では処理R1.dと表す)乱数発生手段を用いて、元の秘密s 及び  $\{1, \dots, p-1\}$  の内にあるランダムに選ばれた秘密の元  $l_1, \dots, l_k$  に対する部分行列を生成し、秘密値  $s_r(1), \dots, s_r(n), l_{1_r}(1), \dots, l_{1_r}(n), \dots, l_{k_r}(1), \dots, l_{k_r}(n)$  に対するハッシュ関数の値  $s^*$  を計算する(図7参照)。

【0086】加入者d は各加入者  $i$  ( $i=1, \dots, n$ 、但し自分は除く)に秘密通信路を利用して、生成した各部分行列の各列ベクトル  $S_c(i), L_{1_c}(i), \dots, L_{k_c}(i)$  及び秘密  $s_r(i), l_{1_r}(i), \dots, l_{k_r}(i)$  (図5では情報B1.i)を秘密通信路を用いて送信し、全加入者間でハッシュ値  $s^*$  (図5では情報B1.d)を、放送通信路を用いて放送する。

【0087】(ラウンド2)各加入者  $i$  ( $i=1, \dots, n$ ) は(図5では処理R2.i)乱数発生手段を用いて、 $k'$  個のランダムに選択されたビットを放送する(図5ではB2.i)。それらのランダムに選ばれた  $k'$  個のビットは  $B_{i_1}, \dots, B_{i_{k'}}$ 、 $n$  人の全ての加入者に対する全ビットは  $B_1, \dots, B_k$  と呼ぶ。

【0088】(ラウンド3)加入者d は(図5では処理R3.d)ラウンド2で放送された各ビット  $B_j$  ( $j=1, \dots, k$ ) に対して、 $B_j$  が1であればラウンド1で加入者d が生成した部分行列  $L_j$  を放送し、 $B_j$  が0であればラウンド1で加入者d が生成した部分行列  $S$  と  $L_j$  の要素毎の有限体上の加算結果 ( $S+L_j$  と書く)を放送する。加入者d が放送した情報は、図5でB3.dと表される。

【0089】(ラウンド4)各加入者  $i$  ( $i=1, \dots, n$ ) は(図5では処理R4.i)ラウンド1で秘密に受信した情報B1.iのなかに、各値  $j$  ( $j=1, \dots, k$ ) に対して列ベクトル  $L_{j_c}(i)$  及び  $l_{j_r}(i)$  (ラウンド2で放送されたビット  $B_j$  が1の場合)あるいは  $L_{j_c}(i)+S_c(i)$  及び  $l_{j_r}(i)+s_r(i)$  ( $B_j$  が0の場合)がラウンド3で放送された部分行列に対応する列ベクトル及び秘密値と等しいかどうかを確認する。ある値  $j$  に対して、等しくなければ、加入者d の判定メッセージ(図5では情報B4.i)を放送する。

【0090】(ラウンド5)加入者d は、ラウンド4で判定メッセージが放送された場合に、判定メッセージを放送した各加入者  $j$  に関してラウンド1で加入者d が秘密に送信した情報B1.j (図5では処理B5.d)を放送する(図5では処理R5.d)。

【0091】(後処理)各加入者  $i$  ( $i=1, \dots, n$ ) は、ラウンド5で放送された情報が正しくなければ、あるいは、ラウンド4で放送された判定メッセージの個数が閾値  $t$  より多ければ、加入者d が不正を働いたと判断する(図5では処理Pi)。ラウンド1~5で各加入者  $i$  が受信した全ての情報を  $s_i$  と書く(図5参照)。

【0092】(2)秘密復元処理(図8参照)：秘密分散処理による各加入者が持っている情報  $s_i$  から、全ての

加入者が元の秘密  $s$  を計算するための処理

(ラウンド1)各加入者  $i$  ( $i=1, \dots, n$ ) は(図8では処理R1.i)部分情報  $s_i$  含まれている秘密値  $s_c(i)$  及び  $s_r(i)$  (図8では情報B1.i)を放送する。

【0093】(ラウンド2)各加入者  $i$  ( $i=1, \dots, n$ ) は(図8では処理R2.i)ラウンド1で放送された値の内  $t+1$  個の値  $s_c(i, 1), \dots, s_c(i, t+1)$  及び  $s_r(i, 1), \dots, s_r(i, t+1)$  を選び、多項式補間処理の結果  $s(c)$  及び  $s(r)$  を求め、両方が等しくなり残りの放送された値がその値  $s(c)=s(r)$  に対応する同じ多項式の正しい値になるかどうかを確認する。全ての値が正しければ元の秘密  $s$  は  $s(c)=s(r)$  と等しいと判断し復元処理が終わる。同じ多項式に対応しない値が放送された場合に、部分情報  $s_i$  に含まれている列ベクトル  $S_c(i)$  を放送する(図8では情報B2.i)。

【0094】(ラウンド3)各加入者  $i$  ( $i=1, \dots, n$ ) は(図8では処理R3.i)ラウンド2で放送された列ベクトル  $S_c(j)$  ( $j=1, \dots, n$ ) から部分行列  $S'$  を生成し、 $1, \dots, n$  より異なる  $t+1$  個の値を含む全ての集合  $t_1, \dots, t_m$  (全部で  $m=n!/((t+1)!(n-t-1)!)$  個の集合がある)に対する列ベクトルの集合  $T_1, \dots, T_m$  を選び、それぞれの行と列に対する多項式補間処理の結果  $s'_r(1), \dots, s'_r(n)$  及び  $s'_c(1), \dots, s'_c(n)$  を求め、更にこれらの値の多項式補間処理の結果  $s'(r)$  及び  $s'(c)$  を求め、両方が等しくなりそれらの列ベクトルの全ての要素がそれぞれの値  $s'_c(1), \dots, s'_c(n)$  及び  $s'_r(1), \dots, s'_r(n)$  に対応する同じ多項式の正しい値になるかどうかを確認する。

【0095】このように確認された  $t+1$  個の列ベクトルを含む集合  $T_1, \dots, T_m$  の内の異なる正しい部分行列に対応する集合の数は最大  $t+1$  個になり、それらの部分行列を  $S_1, \dots, S_T$  (ただし、 $T \leq t+1$ ) と呼ぶ。正しい部分行列が一個しかなければ ( $T=1$ )、その行列に対する秘密  $s$  が元の秘密と等しいと判断し、対応しない列ベクトルが不正をした加入者を表す。正しい部分行列が二つ以上があれば、各加入者  $i$  ( $i=1, \dots, n$ ) は持っている全ての部分情報  $s_i$  を放送する(図8では情報B3.i)。

【0096】(後処理)各加入者  $i$  ( $i=1, \dots, n$ ) は(図8では処理Pi)ラウンド3で放送された情報  $s_j$  ( $j=1, \dots, n$ ) を用いて、ラウンド4で計算した正しい部分行列  $S_1, \dots, S_T$  (ただし、 $T \leq t+1$ ) に対して、秘密分散処理と同じ方向性ハッシュ関数を計算し、全ての放送された情報を正しく、分散処理のラウンド1で放送された値  $s^*$  に対応するかどうかを確認する。これらの確認に対して正しい(最大一つしかありえない)部分行列に対応する秘密  $s'$  は元の秘密  $s$  と等しいと判断する。それに対して前述の確認に対して正しい部分行列がなかったならば、秘密の分散を行った加入者d が不正したと判断する。

【0097】以上により、署名者グループのある加入者



が持っている秘密の元を、そのグループの全加入者間で分散保持する処理が実現できる。次に、この秘密分散方式を用いて、分散デジタル署名方式における署名者グループの秘密情報（秘密鍵に相当する、ただしそのグループに参加する全加入者間で分散されている）、及びその秘密情報に対応する公開情報（公開鍵に相当するそのグループによって生成された署名を確認するための公開情報）を生成する処理について述べる。

【0098】(3) 鍵生成処理（図9参照）

（ラウンド1～5）各加入者 $i$ が $\{1, \dots, p-1\}$ の内に秘密の元 $a(i)$ をランダムに選び、上述の秘密分散処理を利用して署名者グループに参加する全加入者間で分散する。さらに、各加入者 $i$ が $g^{a(i)} \bmod q$ （ただし、 $q$ は前述のように選ばれた素数である）を計算し、放送通信路を利用して放送する。

【0099】（ラウンド6）上述の秘密分散処理の後処理を実行し、正しく分散された秘密の元 $a(i)$ （ $i=1, \dots, n$ ）を入力として以下の分散加算によって得られた分散出力を秘密情報 $a$ とする。さらに、正しく分散された秘密の元 $a(i)$ に対する前の処理のラウンド1～5で放送された値 $A(i)=g^{a(i)} \bmod q$ の乗算を計算し、得られた結果 $A=g^a \bmod q$ をそのグループによる署名を確認するための公開情報（公開鍵）とする。

【0100】ラウンド6で用いた分散加算について図10を参照にして説明する。

【0101】前述の秘密分散処理によって、 $\{1, \dots, p-1\}$ の内に二つの秘密の元 $x$ と $y$ 分散されたとき

（各加入者 $i$ が秘密の元 $x$ と $y$ に対応する秘密部分 $x_i$ 及び $y_i$ 持つ）、通信を行わずに、 $\{1, \dots, p-1\}$ の内に和 $x+y$ に対応する秘密部分 $(x+y)_i$ を次のように計算する。各加入者が持っている部分行列の列ベクトル $X_{c(i)}$ 及び $Y_{c(i)}$ かつ秘密値 $x_r(i)$ 及び $y_r(i)$ の要素毎の加算結果 $X_{c(i)}+Y_{c(i)}$ 及び $x_r(i)+y_r(i)$ が $x+y$ に対する部分行列の列ベクトル及び秘密値になることは、部分行列の定義（行と列の要素が多項式の値となる）から明らかである。

【0102】秘密復元処理において部分情報を確認するために用いられる一方向性ハッシュ関数の値 $x^*$ 及び $y^*$ は、両方を記憶し、加算結果 $x+y$ を復元処理のとき必要であれば、両方を用いることによって秘密 $x+y$ に対応する部分行列 $X+Y$ を確認する。同じように、分散された一つの秘密の元 $x$ と公開の元 $a$ との乗算を分散して計算するために、各加入者が持っている部分行列の各要素と $a$ との乗算を行った結果が $x^*a$ に対する部分行列の要素になる。

【0103】よって、上述の処理によって分散された二つの秘密の元 $x$ と $y$ 及び公開の元 $a$ と $b$ の線形結合 $a^*x+b^*y$ はそのグループに参加する加入者間で対話せずに分散して計算できる。この線形結合処理を図11のように表現する。ただし、図は行う処理の名前と、その処理に

対する各加入者毎の入出力を示している。各加入者 $i$ （ $i=1, \dots, n$ ）の入力は、分散された秘密の元 $x$ と $y$ に対する部分情報 $x_i$ と $y_i$ 及び公開の元 $a$ と $b$ になり、出力は線形結合処理の結果 $z$ に対する部分情報 $z_i$ になる。

【0104】以上の処理によって得られた秘密鍵を利用して、与えられたメッセージ $m$ のデジタル署名がそのグループによって次のように分散して生成される。

【0105】(4) 署名生成処理（図12参照）

（ラウンド1～5）各加入者 $i$ が $\{1, \dots, p-1\}$ の内に秘密の元 $r(i)$ をランダムに選び、上述の秘密分散処理を利用して署名者グループに参加する全加入者間で分散する。さらに、各加入者 $i$ が $g^{r(i)} \bmod q$ （ただし、 $q$ は前述のように選ばれた素数である）を計算し、放送通信路を利用して放送する。

【0106】（ラウンド6）上述の秘密分散処理の後処理を実行し、正しく分散された秘密の元 $r(i)$ （ $i=1, \dots, n$ ）を入力として先述の分散加算によって分散結果 $r$ を求める。さらに、正しく分散された秘密の元 $r(i)$ に対するラウンド1～5で放送された値 $R(i)=g^{r(i)} \bmod q$ の乗算 $R=g^r \bmod q$ を計算する。

【0107】（ラウンド7）各加入者は与えられたメッセージ $m$ とラウンド6で計算された値 $R$ を連続した値 $R|m$ を入力として前述の所定の関数 $h$ の出力 $e=h(R|m)$ を計算し、署名者グループに参加する全加入者間で上述の分散線形結合処理を利用して、 $s=r+h(R|m)^*a$ を分散して計算する。

【0108】（ラウンド8～10）上述の秘密分散方式の秘密復元処理を利用して、分散秘密 $s$ を復元する。与えられたメッセージの署名は $(R, s)$ とする。公開鍵 $a$ 及び前述のデジタル署名方式の署名確認処理を利用して、生成された署名を確認し、正しくないとき不正をした加入者が存在すると判断し、前述の秘密分散方式の秘密復元処理を実行することによって不正した加入者を識別する。

【0109】【実施例2】この実施例では前述のA. Fiat, A. Shamirによって提案されたデジタル署名方式を用いた分散デジタル署名方式を実現する具体的な構成について述べる。まず、 $\{1, \dots, N-1\}$ の内に秘密の元を確認可能に分散する具体的な構成を次に示す。

【0110】ある与えられた秘密 $s$ に対する部分行列の具体的な説明を行う（図4参照）。 $\{1, \dots, N-1\}$ の内に秘密の元 $s$ に対する部分行列 $S=[s(i, j)]$ ,  $i, j=1, \dots, n$ とは、各行ベクトル $(S_r(i)=[s(i, 1), \dots, s(i, n)]$ ,  $i=1, \dots, n$ )の要素が次のように定義されるものである。

【0111】 $s(i, j)=s_r(i)*q_r(i, 1)^{(j)}*q_r(i, 2)^{(j^2)}*\dots*q_r(i, t)^{(j^t)} \bmod N$  ( $j=1, \dots, n$ )

ただし、 $q_r(i, 1), \dots, q_r(i, t)$ は $\{1, \dots, N-1\}$ の内で以下の条件が満たされるように選ばれた元である。各

列ベクトル  $(S_c(j) = [s(1, j), \dots, s(n, j)]$ ,  $j=1, \dots, n$ ) の要素は次のように定義される。

【0112】  $s(j, i) = s_c(j) * q_c(j, 1)^{(i)} * q_c(j, 2)^{(i^2)} * \dots * q_c(j, t)^{(i^t)} \bmod N$  ( $i=1, \dots, n$ )

ただし、 $q_c(j, 1), \dots, q_c(j, t)$  は  $\{1, \dots, N-1\}$  の内で以上の条件が満たされるように選ばれた元である。さらに前述の値のベクトル  $[s_r(1), \dots, s_r(n)]$  及び  $[s_c(1), \dots, s_c(n)]$  の両方は、 $\{1, \dots, N-1\}$  の内にある  $q_r(1), \dots, q_r(t)$ 、 $q_c(1), \dots, q_c(t)$  に対して次の条件を満たす。

【0113】  $s_r(j) = s * q_r(1)^{(j)} * q_r(2)^{(j^2)} * \dots * q_r(t)^{(j^t)} \bmod N$  ( $j=1, \dots, n$ )

$s_c(i) = s * q_c(1)^{(i)} * q_c(2)^{(i^2)} * \dots * q_c(t)^{(i^t)} \bmod N$  ( $i=1, \dots, n$ )

ただし、 $s$  は元の秘密を表す。

【0114】秘密の元  $s$  を署名者グループに参加する全加入者間で秘密に分散保持されるように秘密部分を分配する秘密分散処理と、そのように分散された秘密または（不正があった場合に）不正をした加入者を明かにする秘密復元処理の二つの処理が前述の実施例 1 の処理 (1)、(2) のように行われる。ただし、秘密復元処理 (2) において行われる多項式補間処理の代わりに、各行ベクトル  $(S_r(i) = [s(i, 1), \dots, s(i, n)]$ ,  $i=1, \dots, n$ ) の内にある  $t+1$  個の要素  $(s(i, j_0), \dots, s(i, j_t))$  から値  $s_r(i)$  を求めるために、次の計算が行われる。

【0115】  $s_r(i)^{(n!)} = \text{Prod}_k(s(i, j_k)^{(\text{Prod}_l(l * n! / (l - k)))) \bmod N$

ただし、 $\text{Prod}_k(f(k))$  は  $k=j_0, \dots, j_t$  に対する値  $f(k)$  の乗算を表し、 $\text{Prod}_l(g(l))$  は  $l=j_0, \dots, j_t$  ( $l \neq k$ ) に対する値  $g(l)$  の乗算を表す。

【0116】同じように、各列ベクトル  $(S_c(i) = [s(1, i), \dots, s(n, i)]$ ,  $i=1, \dots, n$ ) の内にある  $t+1$  個の要素  $(s(i_0, j), \dots, s(i_t, j))$  から値  $s_c(j)$  を求めるために、次の計算が行われる。

【0117】  $s_c(j)^{(n!)} = \text{Prod}_k(s(i_k, j)^{(\text{Prod}_l(l * n! / (l - k)))) \bmod N$

これらの部分ベクトル  $[s_r(1)^{(n!)}, \dots, s_r(n)^{(n!)}]$  (あるいは  $[s_c(1)^{(n!)}, \dots, s_c(n)^{(n!)}]$ ) の内にある  $t+1$  個の要素  $(s_r(i_0)^{(n!)}, \dots, s_r(i_t)^{(n!)})$ 、あるいは  $(s_c(j_0)^{(n!)}, \dots, s_c(j_t)^{(n!)})$  から秘密  $s$  を求めるために次の計算が行われる ( $s_r(i)^{(n!)}$  の場合)。

【0118】  $s^{(n! * n!)} = \text{Prod}_k(s_r(k)^{(n! * \text{Prod}_l(l * n! / (l - k)))) \bmod N$

$s = (s^{(n! * n!)})^{u * (s^l)^v} \bmod N$

ただし、A. Fiat, A. Shamir によるデジタル署名方式に用いられる  $l$  は、 $u * n! * n! + v * l = 1$  になるような  $u, v$  が存在するように選ばれる。分散された秘密  $s$  に対する値  $s^l \bmod N$  は、後に (6) において説明する署名処理において計算されている。

【0119】以上により、署名者グループのある加入者が持っている秘密の元をそのグループの全加入者間で分散保持する処理が実現できる。次に、この秘密分散方式を用いて分散デジタル署名方式における署名者グループの秘密情報（秘密鍵に相当する、ただしそのグループに参加する全加入者間で分散されている）、及びその秘密情報に対応する公開情報（公開鍵に相当するそのグループによって生成された署名を確認するための公開情報）を生成する処理について述べる。

【0120】(5) 鍵生成処理 (図 13 参照)

(ラウンド 1 ~ 5) 各加入者  $i$  が  $\{1, \dots, N-1\}$  の内にある秘密の元  $a(i)$  をランダムに選び、上述の秘密分散処理を利用して署名者グループに参加する全加入者間で分散する。さらに、各加入者  $i$  が  $a(i)^l \bmod N$  (ただし、 $l$  は前述のように選ばれた元である) を計算し、放送通信路を利用して放送する。

【0121】(ラウンド 6) 上述の秘密分散処理の後処理を実行し、正しく分散された秘密の元  $a(i)$  ( $i=1, \dots, n$ ) を入力として以下の分散乗算によって得られた分散出力を秘密情報  $a$  とする。さらに、正しく分散された秘密の元  $a(i)$  に対する前の処理のラウンド 1 ~ 5 で放送された値  $A(i) = a(i)^l \bmod N$  の乗算を計算し、得られた結果  $A = a^l \bmod N$  をそのグループによる署名を確認するための公開情報（公開鍵）とする。

【0122】ラウンド 6 で用いた分散乗算について図 14 を参照にして説明する。

【0123】前述の秘密分散処理によって、 $\{1, \dots, N-1\}$  の内にある二つの秘密の元  $x$  と  $y$  が分散されたとき（各加入者  $i$  が秘密の元  $x$  と  $y$  に対応する秘密部分  $x_i$  及び  $y_i$  持つ）、通信を行わずに、 $\{1, \dots, N-1\}$  の内にある積  $x * y$  に対応する秘密部分  $(x * y)_i$  を次のように計算する。

【0124】各加入者が持っている部分行列の列ベクトル  $X_c(i)$  及び  $Y_c(i)$  かつ秘密値  $x_r(i)$  及び  $y_r(i)$  の要素毎の乗算結果  $X_c(i) * Y_c(i)$  及び  $x_r(i) * y_r(i)$  が  $x * y$  に対する部分行列の列ベクトル及び秘密値になることは、部分行列の定義から明らかである。秘密復元処理において部分情報を確認するために用いられる一方向性ハッシュ関数の値  $x^*$  及び  $y^*$  は、両方を記憶し、乗算結果  $x * y$  を復元処理のとき必要であれば、両方を用いることによって秘密  $x * y$  に対応する部分行列  $X * Y$  を確認する。

【0125】同じように、分散された一つの秘密の元  $x$  と公開の元  $a$  とのべき乗算を分散して計算するために、各加入者が持っている部分行列の各要素と  $a$  とのべき乗算を行った結果が  $x^* a$  に対する部分行列の要素になる。よって、上述の処理によって分散された二つの秘密の元  $x$  と  $y$  及び公開の元  $a$  の結合  $x^* a * y^* b$  はそのグループに参加する加入者間で対話せずに分散して計算できる。これを図 11 と同様に図 15 のように表す。

【0126】以上の処理によって得られた秘密鍵を利用

【0140】(ラウンド2)各加入者 $i$  ( $i=1, \dots, n$ )  
(図17では処理R2.i)は、ラウンド1で受け取った秘密 $s_r(i)$ ,  $l1_r(i)$ ,  $\dots$ ,  $lk_r(i)$ 及び以前分散された $x$ に対する秘密 $x_r(i)$ を入力として、 $t_r(i)=s_r(i)*x_r(i)$ ,  $m1_r(i)=l1_r(i)*x_r(i)$ ,  $\dots$ ,  $mk_r(i)=lk_r(i)*x_r(i) \bmod N$ の計算を行い、乱数発生手段を用いて、得られた結果に対する部分行列  $(T(i), M1(i), \dots, Mk(i))$  と呼び、

以前分散された秘密 $x$ の部分 $x_r(i)$ に対する部分行列を $X(i)$ と呼ぶ)を生成し、秘密値 $t(i)_r(1), \dots, t(i)_r(n), m1(i)_r(1), \dots, m1(i)_r(n), \dots, mk(i)_r(1), \dots, mk(i)_r(n)$ に対するハッシュ関数の値 $s^*$ を計算する(図7参照)。各加入者 $i$ は各加入者 $j$  ( $j=1, \dots, n$ 、ただし自分は除く)に秘密通信路を利用して、生成した各部分行列の各列ベクトル $T(i)_c(j), M1(i)_c(j), \dots, Mk(i)_c(j)$ 及び秘密 $t(i)_r(j), m1(i)_r(j), \dots, mk(i)_r(j)$ を秘密通信路を用いて送信し、全加入者間でハッシュ値 $s^*$ を、放送通信路を用いて放送する。加入者 $i$ が放送した情報は、図17でB2.iと表される。

【0141】(ラウンド3)各加入者 $j$  ( $j=1, \dots, n$ ) (図17では処理R3.i)は、乱数発生手段を用いて、 $k'$ 個のランダムに選択されたビットを放送する。それらのランダムに選ばれた $k'$ 個のビットを $Bj_1, \dots, Bj_{k'}$ 、 $n$ 人の全ての加入者に対する全ビットを $B1, \dots, Bk$ と呼ぶ。加入者 $i$ が放送した情報は、図17でB3.iと現される。

【0142】(ラウンド4)加入者 $d$  (図17では処理R4.d)は、ラウンド3で放送された各ビット $Bj$  ( $j=1, \dots, k$ )に対して、 $Bj$ が1であればラウンド1で加入者 $d$ が生成した部分行列 $Lj$ を放送し、 $Bj$ が0であればラウンド1で加入者 $d$ が生成した部分行列 $S$ と $Lj$ の要素毎の有体上(有限体)の加算結果( $S+Lj$ と書く)を放送する(図17でB4.d)。

【0143】(ラウンド5)各加入者 $i$  ( $i=1, \dots, n$ ) (図17では処理R5.i)は、ラウンド1で秘密に受信した情報 $B1.i$ の中の $j$  ( $j=1, \dots, k$ )に対する各列ベクトル $Lj_c(i)$ 及び $lj_r(i)$  (ラウンド2で放送されたビット $Bj$ が1の場合)あるいは $Lj_c(i)+S_c(i)$ 及び $lj_r(i)+s_r(i)$  ( $Bj$ が0の場合)がラウンド4で放送された部分行列に対応する列ベクトル及び秘密値と等しいかどうかを確認する。ある値 $j$ に対して等しくなければ加入者 $d$ の判定メッセージを放送する。加入者 $i$ が放送した情報は、図17、18でB5.iと表される。

【0144】(ラウンド6)各加入者 $i$  (図18では処理R6.i)は、ラウンド3で放送された各ビット $Bj$  ( $j=1, \dots, k$ )に対して、 $Bj$ が1であればラウンド2で加入者 $i$ が生成した部分行列 $Mj(i)$ を放送し、 $Bj$ が0であればラウンド1で加入者 $d$ が生成した部分行列 $T(i)$ と $Mj(i)$ の要素毎の有体上(mod  $p$ )の加算結果( $T(i)+Mj(i)$ と書く)を放送する。更に、加入者 $d$ は、ラウンド5で判定した加入者の列ベクトルを放送する。加入者 $i$ が放送した情報は、図18でB6.iと表される。

【0145】(ラウンド7)各 $j$ に対して( $j=1, \dots, k$ )、各加入者 $i$  (図18では処理R7.i)は、ラウンド5~6で放送された情報を確認する。しきい値 $t$ より多くの行列が正しく、かつ、ラウンド3で放送された各ビット $Bj$  ( $j=1, \dots, k$ )に対して、 $Bj$ が1であれば次の部分行列を放送する。

【0146】 $(lj_r(i))^{(-1)} * Mj(i) - X(i)$

また、 $Bj$ が0であれば次の部分行列を放送する。

【0147】

$(s_r(i)+lj_r(i))^{(-1)} * (T(i)+Mj(i)) - X(i)$

ただし、部分行列の線形結合は前述のように(要素毎)に行われる。加入者 $i$ が放送した情報は、図18でB7.iと現される。

【0148】(ラウンド8)各 $j, o$ に対して( $j, o=1, \dots, k$ )、各加入者 $i$  ( $i=1, \dots, n$ ) (図18では処理R8.i)は、ラウンド4で放送された情報を利用して、 $Bo$ が1であれば $lj_r(o)$ を復元した後に次の列ベクトルを計算し、列ベクトル結果が正しくなること、及び値0に対応することを確認する。

【0149】

$(lj_r(o))^{(-1)} * Mj(o)_r(i) - X(o)_r(i)$

$Bo$ が0であれば $(s_r(o)+lj_r(o))$ を復元した後に次の列ベクトルを計算し、列ベクトル結果が正しくなること、及び値0に対応することを確認する。

【0150】 $(s_r(o)+lj_r(o))^{(-1)} * (T(o)_r(i)+Mj(o)_r(i)) - X(o)_r(i)$

ある値 $o$ に対して確認できなければ加入者 $o$ の判定メッセージを放送する。加入者 $i$ が放送した情報は、図18でB8.iと現される。

【0151】(ラウンド9)各加入者 $o$  (図18では処理R9.o)は、ラウンド8で自分に対して判定メッセージが放送された場合に、その列ベクトルを放送する。加入者 $i$ が放送した情報は、図18でB9.iと現される。

【0152】(ラウンド10)各加入者 $i$  (図18では処理R10.i)は、全ての放送された情報を確認し、不正を行った他の加入者 $o$ に対して判定メッセージを放送する。加入者 $i$ が放送した情報は、図18でB10.iと現される。

【0153】(後処理)全加入者によって(各加入者 $i$ の処理が図18では処理Piで表される)正しく分散された部分行列 $S(i)*X(i)$ の分散線形結合を行うことによって正しい部分行列 $S*X$ を分散して計算する。不正によってその行列を生成できない場合には、不正を行った加入者を識別し、その結果を放送する。

【0154】以上の処理を行うことによって、ある加入者 $i$ がランダムに選んだ $\{1, \dots, p-1$ の内にある秘密の元 $s$ は、確認可能な形で分散が行われながら、同時にその秘密 $s$ と以前に確認可能に分散された秘密 $x$ との積 $s*x$ が計算される。次に、上述の処理を利用して構成される二つの分散されている秘密 $x$ と $y$ の積を実現する計算についてのべる。これを図19に示す。

【0155】(ラウンド1~10)各加入者 $i$ がランダムに $\{1, \dots, p-1$ の内にある秘密の元 $r(i)$ を選択し、上述の処理によって確認可能な分散を行いながら $r(i)*y$ を同時に計算する。

【0156】(ラウンド11)正しく確認可能な分散が

行われている全ての加入者  $j$  ( $i=1, \dots, m$ ) に対する秘密の加算  $r = r(j_1) + r(j_2) + \dots + r(j_m)$  と  $r * y = r(j_1) * y + r(j_2) * y + \dots + r(j_m) * y$  が前述の分散加算によって計算される。

【0157】(ラウンド12) 秘密の加算  $u = x - r$  が分散加算される。

【0158】(ラウンド13~15) 全ての加入者によって、秘密  $u$  が復元される。

【0159】(ラウンド16) 秘密の線形結合  $z = r * y - u * y = x * y$  が分散計算される。

【0160】上述の処理によって分散された二つの秘密の元  $x$  と  $y$  の積  $x * y$  はそのグループに参加する加入者間で分散して計算できる。以上の処理によって得られた秘密鍵を利用して、与えられたメッセージ  $m$  のデジタル署名がそのグループによって次のように分散して生成される。

【0161】(8) 署名生成処理 (図20参照)

(ラウンド1~5) 各加入者  $i$  が  $\{1, \dots, p-1\}$  の内にある秘密の元  $r(i)$  をランダムに選び、上述の秘密分散処理を利用して署名者グループに参加する全加入者間で分散する。さらに、各加入者  $i$  が  $g^{r(i)} \bmod q$  (ただし、 $q$  は前述のように選ばれた素数である) を計算し、放送通信路を利用して放送する。

【0162】(ラウンド6) 上述の秘密分散処理の後処理を実行し、正しく分散された秘密の元  $r(i)$  ( $i=1, \dots, n$ ) を入力として先述の分散加算によって分散結果  $r$  を求める。さらに、正しく分散された秘密の元  $r(i)$  に対するラウンド1~5で放送された値  $R(i) = g^{r(i)} \bmod q$  の乗算  $R = g^r \bmod q$  を計算する。

【0163】(ラウンド7) 各加入者は与えられたメッセージ  $m$  を入力として前述の所定の関数  $h$  の出力  $e = h(m)$  を計算し、署名者グループに参加する全加入者間で上述の分散線形結合処理を利用して、 $b = (e + R * a)$  を分散して計算する。

【0164】(ラウンド8~17) 署名者グループに参加する全加入者間で上述の分散乗算処理を利用して、 $s = b * r^{-1}$  を分散して計算する。ただし、 $r^{-1}$  の分散逆元処理を避けるためには、乱数生成処理 (ラウンド1~5参照) において  $g^{r(i)} \bmod q$  の代わりに  $g^{((r(i))^{-1})} \bmod q$  を計算すればよい。

【0165】(ラウンド18~20) 上述の秘密分散方式の秘密復元処理を利用して、分散秘密  $s$  を復元する。与えられたメッセージの署名は  $(R, s)$  とする。公開鍵  $a$  及び前述のデジタル署名方式の署名確認処理を利用して、生成された署名を確認し、正しくないとき不正をした加入者が存在すると判断し、前述の秘密分散方式の秘密復元処理を実行することによって不正した加入者を識別する。

【0166】〔他の実施例〕本発明は、実施例1~3に示した部分行列の次元数等に限定されるものではなく、

さらに多次元の部分配列でもよく、用いる関数は一方向性ハッシュ関数でなくても、一方向性が保証されれば他の関数でもよい。また、Cut and Choose技術も実施例1に具体的に示した手順でなくても、全ての秘密を漏らすことなくその正当性を確認できる手法であればよい。

【0167】以上説明したように、本実施例では、通信システムによって接続されている複数の計算装置 (署名者) からなる署名者グループによって、そのグループの中のあるしきい値  $t$  より多くの署名者が正しければ署名を生成でき、不正があった場合には不正をした署名者を識別できるという意味の分散デジタル署名方式を前述の通り構成できる。この方式は従来技術による不正があっても正しい署名を生成できるという分散デジタル署名方式に比べて、通信量と計算量がより効率的になっている。

【0168】また、従来技術による不正があった場合に不正者が識別できず署名も生成できないという分散デジタル署名方式に対しては、不正を行った署名者を識別できるという点において、本発明による方式の方が安全であると言える。

【0169】具体的に、ある署名者グループに参加する各署名者に必要な計算量は、一番計算量がかかる処理がべき乗演算となるので、従来の後者のデジタル署名方式においてある署名者に必要な計算量とほぼ同じと考えられる。しかし、各署名者に必要な通信量は、 $l * n^2 * k$  のオーダー ( $n$  は加入者の数、 $k$  は安全パラメータ、 $l$  は用いられる整数の長さ) になり、従来の前者のデジタル署名方式に比べて実用的であると考えられる。

【0170】ただし、秘密復元処理 (前述の実施例1の処理(2)のラウンド3参照) で行う  $m = n! / ((t+1)!(n-t-1)!)$  の異なる  $t+1$  列ベクトルを含む集合の数によって、 $n$  は十分小さく ( $n < 20$ ) なければ、不正を行った加入者の識別は実用上で困難であると考えられる。よって、本実施例による秘密分散方式は加入者の数が小さい場合に対して効果的であると言える。しかし、従来の確認可能な秘密分散方式は加入者の数が小さい場合に対しても非実現的な通信量と計算量であった。

【0171】

【発明の効果】以上説明したように、本発明による署名生成方法を用いることによって通信量及び計算量が削減できる。また、これにより、通信量が少ないことによって通信システムのトラフィックや通信料金等が改善され、計算量が少ないことによって処理が高速化されるという効果を生じる。

【図面の簡単な説明】

【図1】本発明の1実施例の通信システムの機能構成を示すブロック図である。

【図2】情報処理装置のブロック構成を示す図である。

【図3】確認可能な秘密分散方式の手順を示す図である。

【図 4】秘密部分行列の説明図である。

【図 5】秘密分散処理手順の説明図である。

【図 6】一方向性ハッシュ関数の具体例を示す図である。

【図 7】一方向性ハッシュ関数の説明図である。

【図 8】秘密復元処理手順の説明図である。

【図 9】秘密鍵及び公開鍵生成処理手順の説明図である。

【図 10】分散秘密の加算処理の説明図である。

【図 11】分散秘密の線形結合処理の入出力関係の説明図である。

【図 12】分散署名生成処理手順の説明図である。

【図 13】秘密鍵及び公開鍵生成処理手順の説明図である。

【図 14】分散秘密の乗算処理の説明図である。

【図 15】分散秘密の結合処理の入出力関係の説明図である。

【図 16】分散署名生成処理手順の説明図である。

【図 17】確認可能な秘密及び分散乗算処理手順の説明図である。

【図 18】確認可能な秘密及び分散乗算処理手順の説明図である。

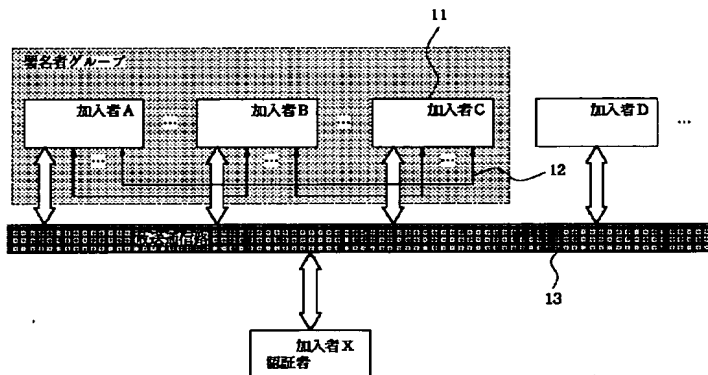
【図 19】分散秘密の乗算処理手順の説明図である。

【図 20】分散署名生成処理手順の説明図である。

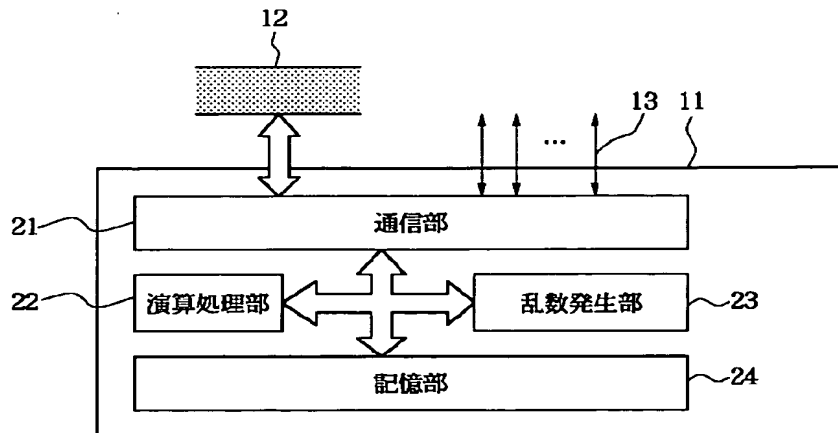
【符号の説明】

- 11 情報処理装置
- 12 放送通信路
- 13 秘密通信路
- 21 通信部
- 22 演算処理部
- 23 乱数発生部
- 24 記憶部
- 61 暗号化回路
- 62 関数演算回路
- 63 ハッシュ演算処理部

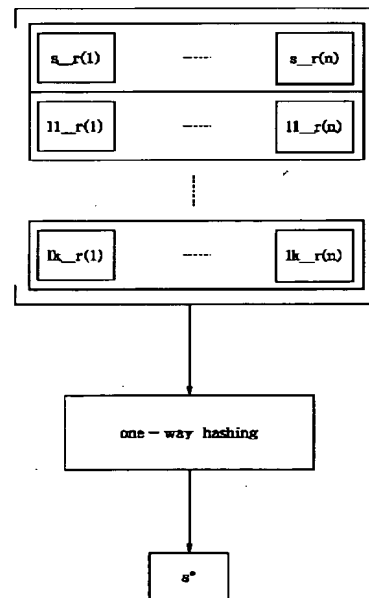
【図 1】



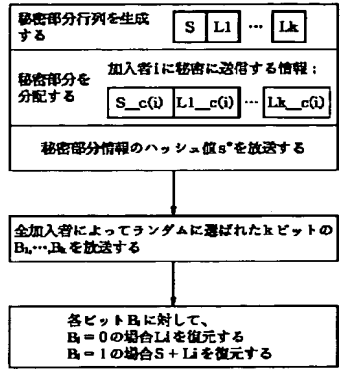
【図 2】



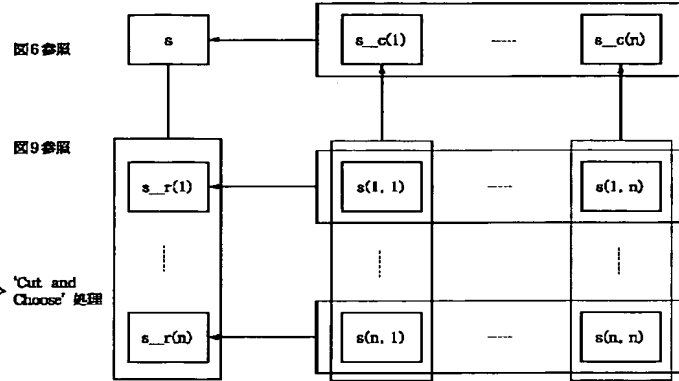
【図 7】



【図3】



【図4】



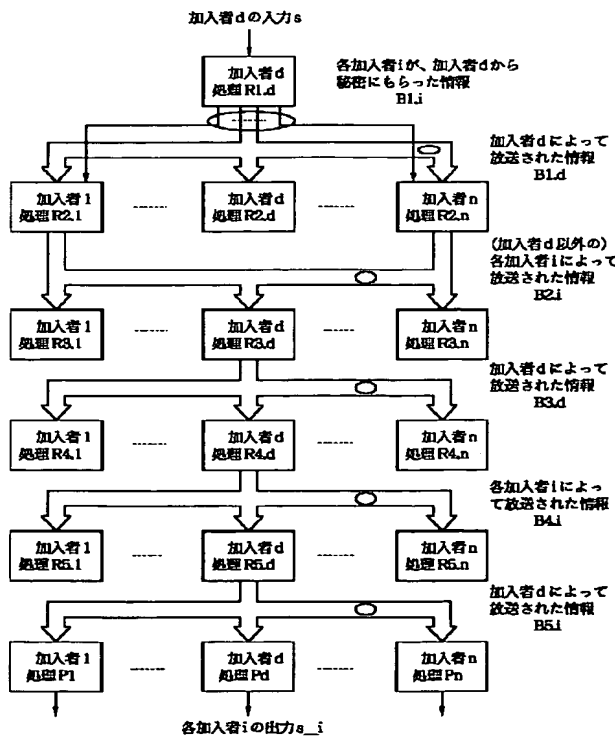
$$S = \begin{bmatrix} s(1, 1) & \dots & s(1, n) \\ \vdots & & \vdots \\ s(n, 1) & \dots & s(n, n) \end{bmatrix}$$

$$S_{-c}(1) = \begin{bmatrix} s(1, 1) \\ \vdots \\ s(n, 1) \end{bmatrix} \dots S_{-c}(n) = \begin{bmatrix} s(1, n) \\ \vdots \\ s(n, n) \end{bmatrix}$$

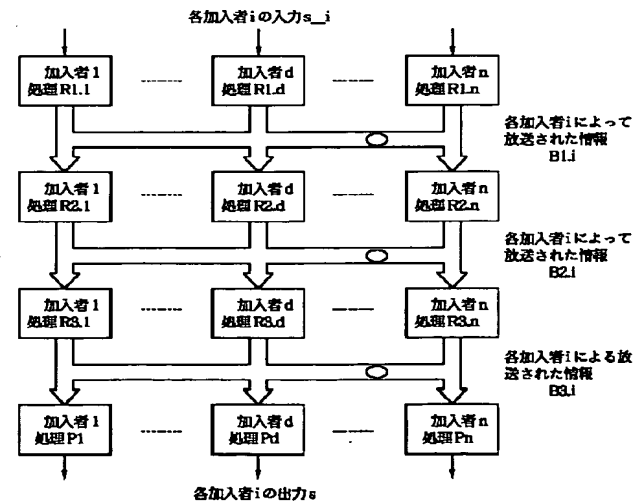
$$S_{-r}(1) = [s(1, 1) \dots s(1, n)]$$

$$S_{-r}(n) = [s(n, 1) \dots s(n, n)]$$

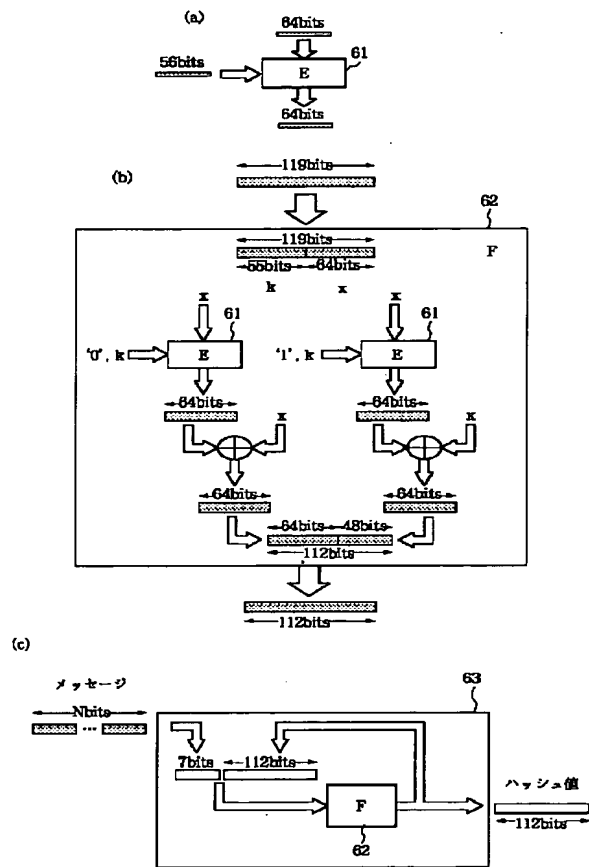
【図5】



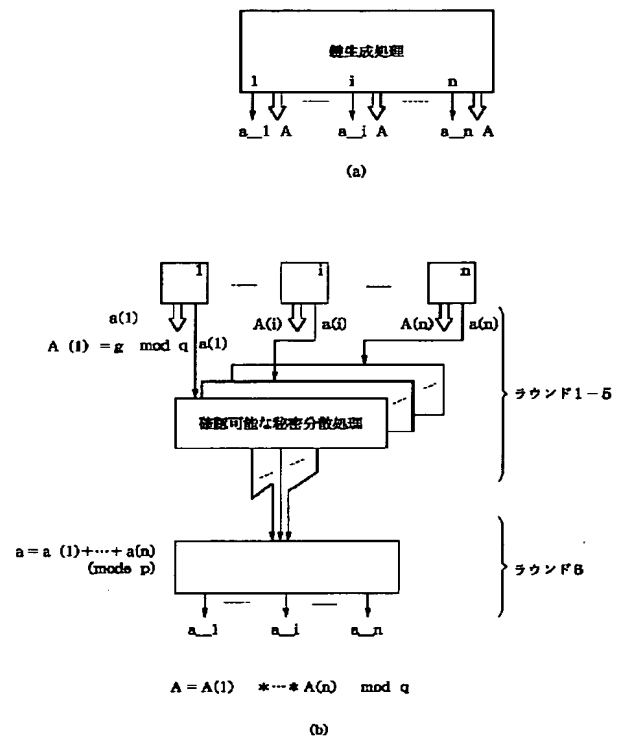
【図8】



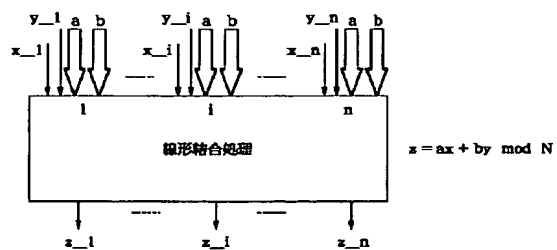
【図 6】



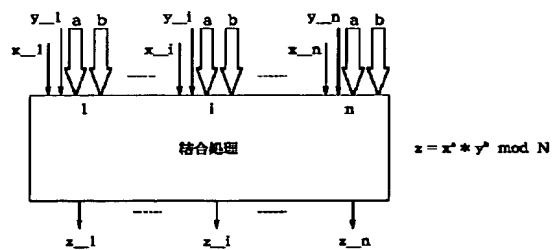
【図 9】



【図 11】

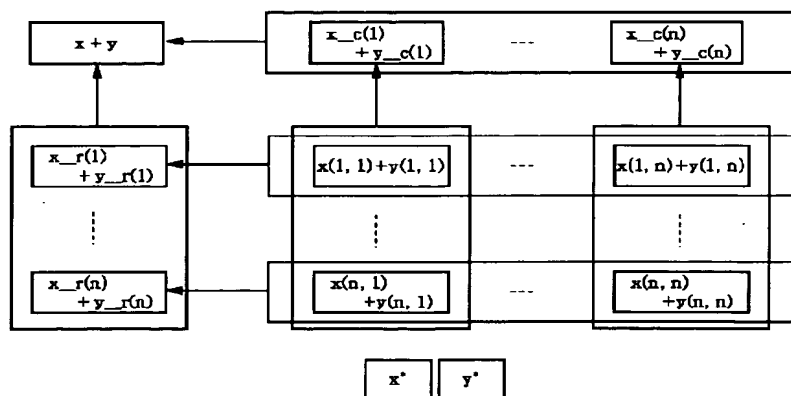
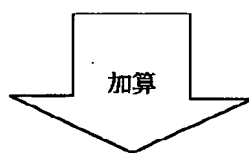
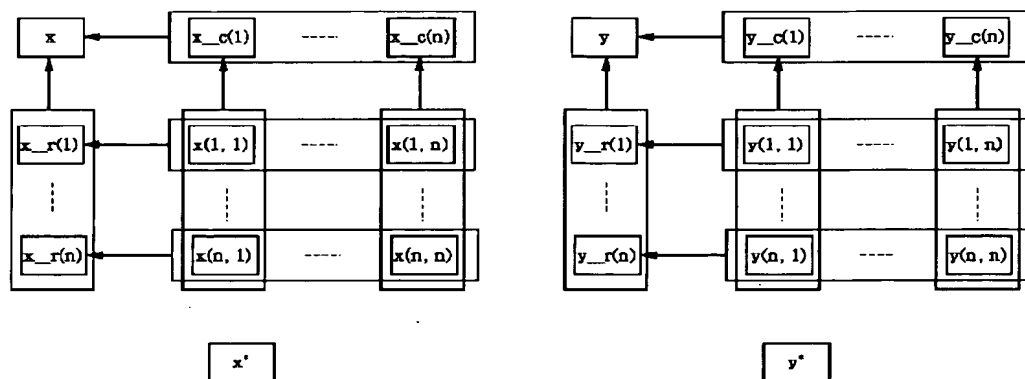


【図 15】

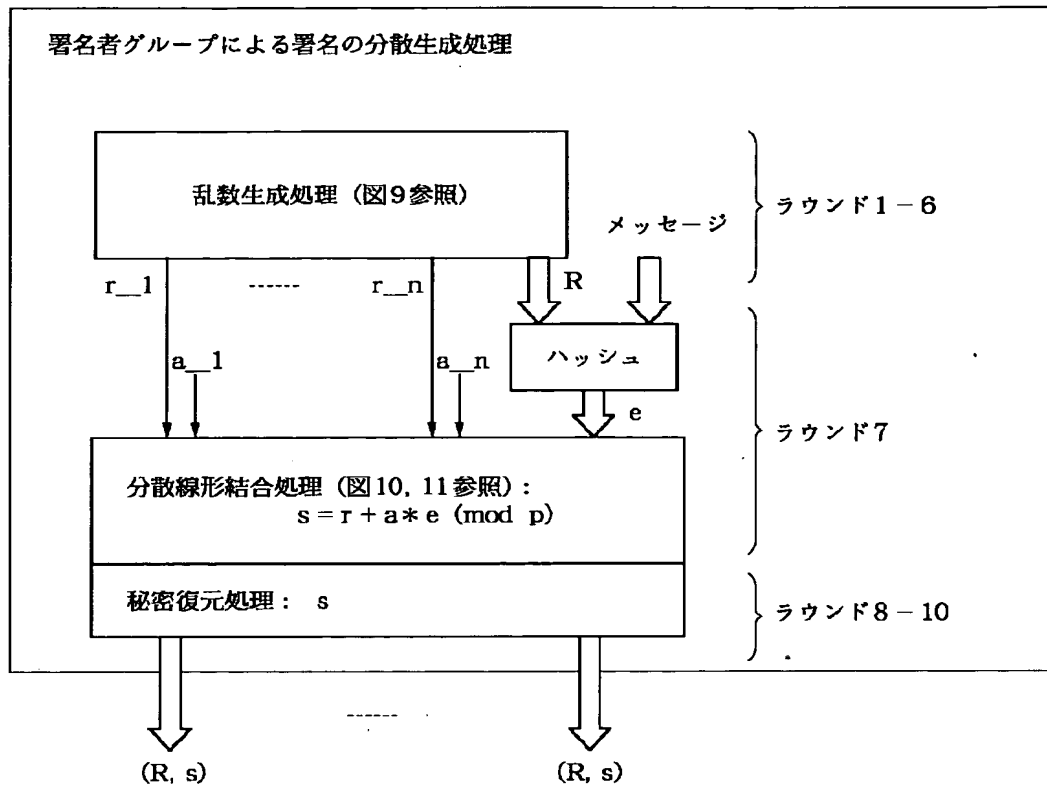




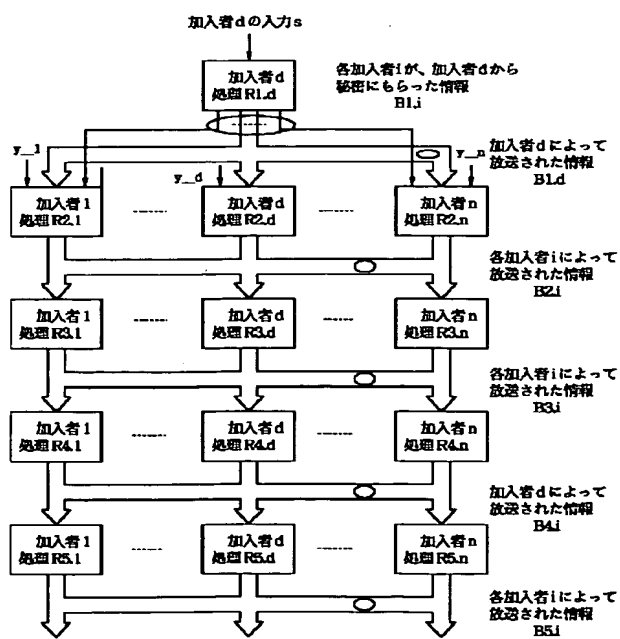
【図10】



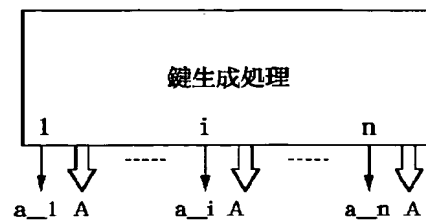
【図12】



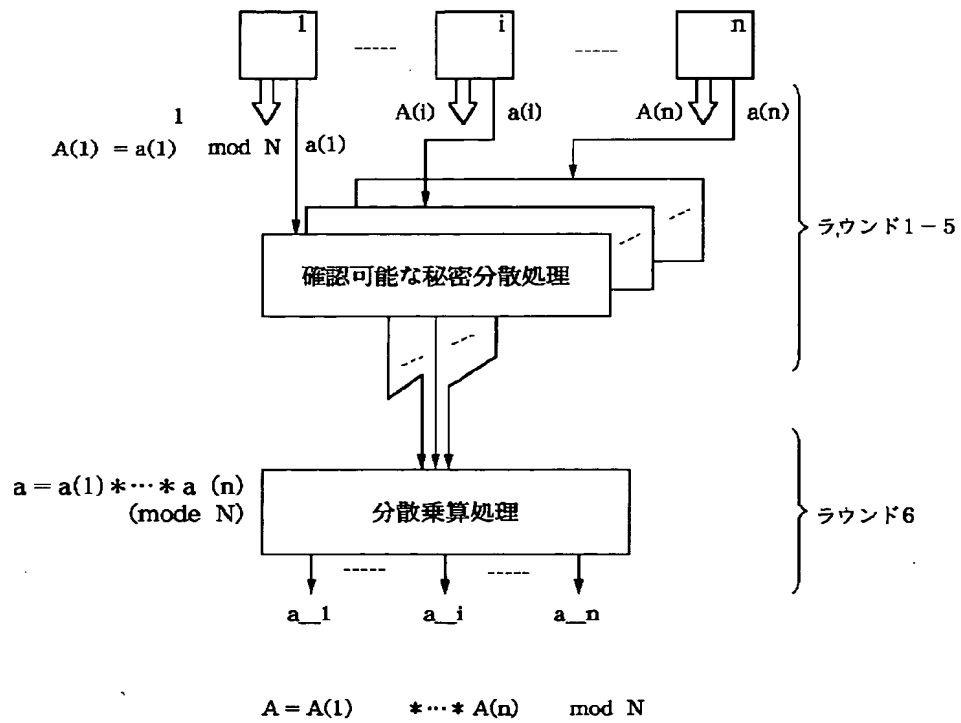
【図17】



【図13】

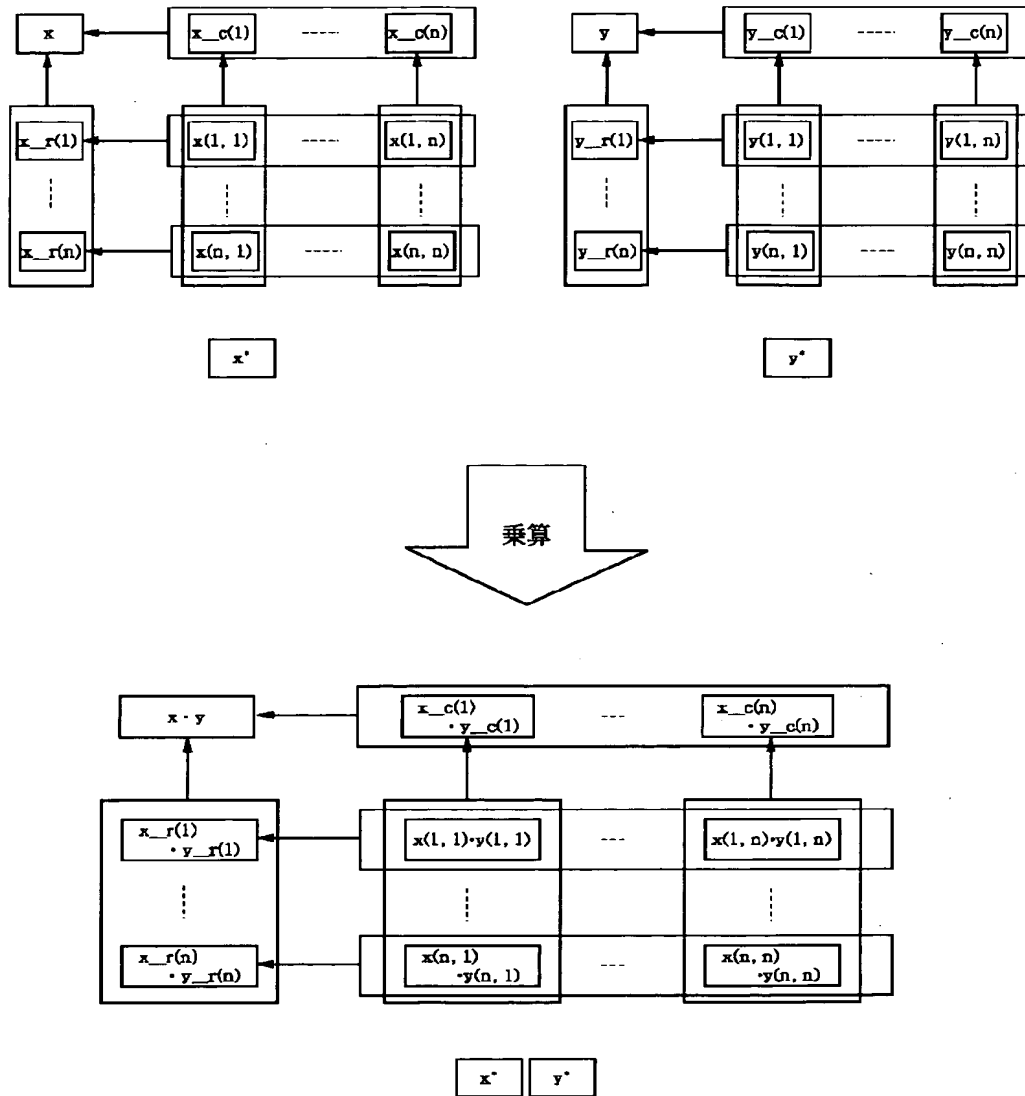


(a)

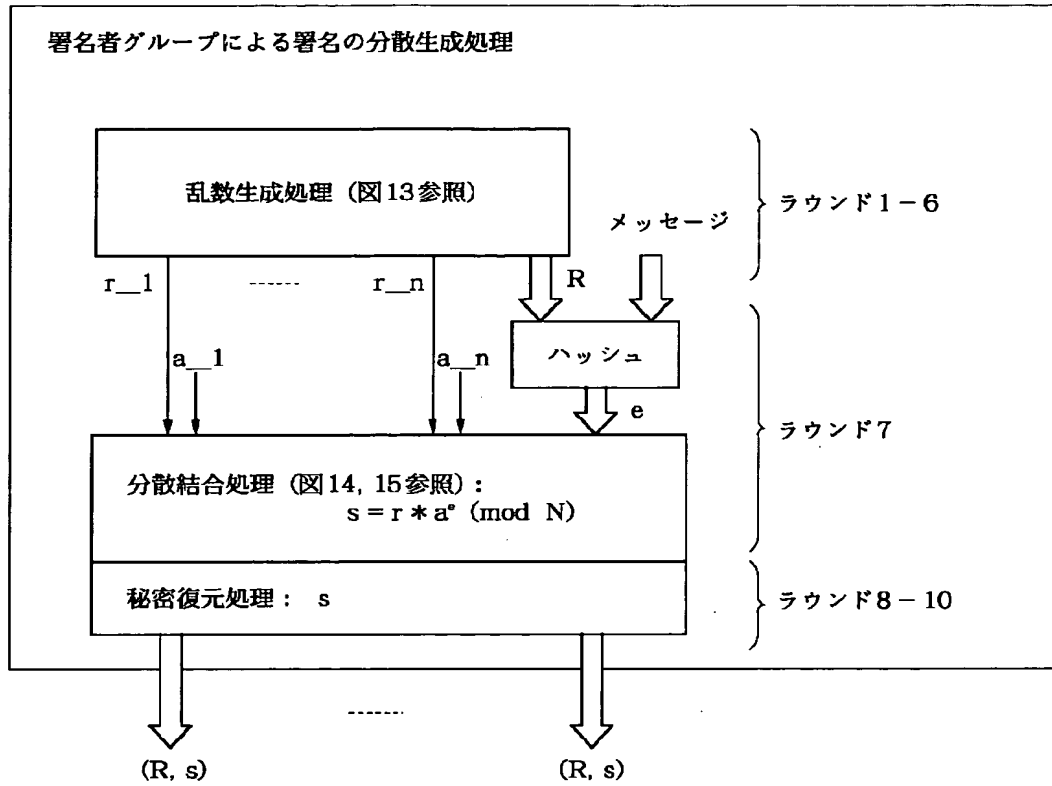


(b)

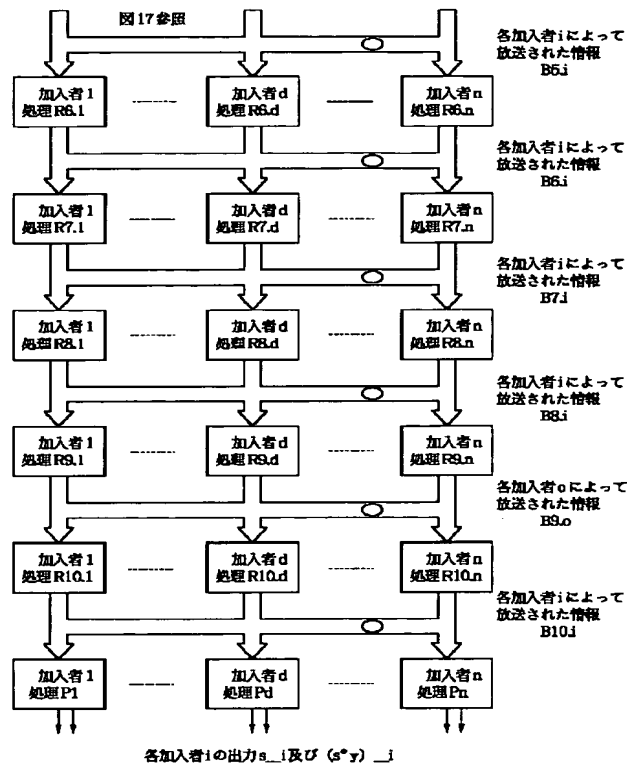
【図14】



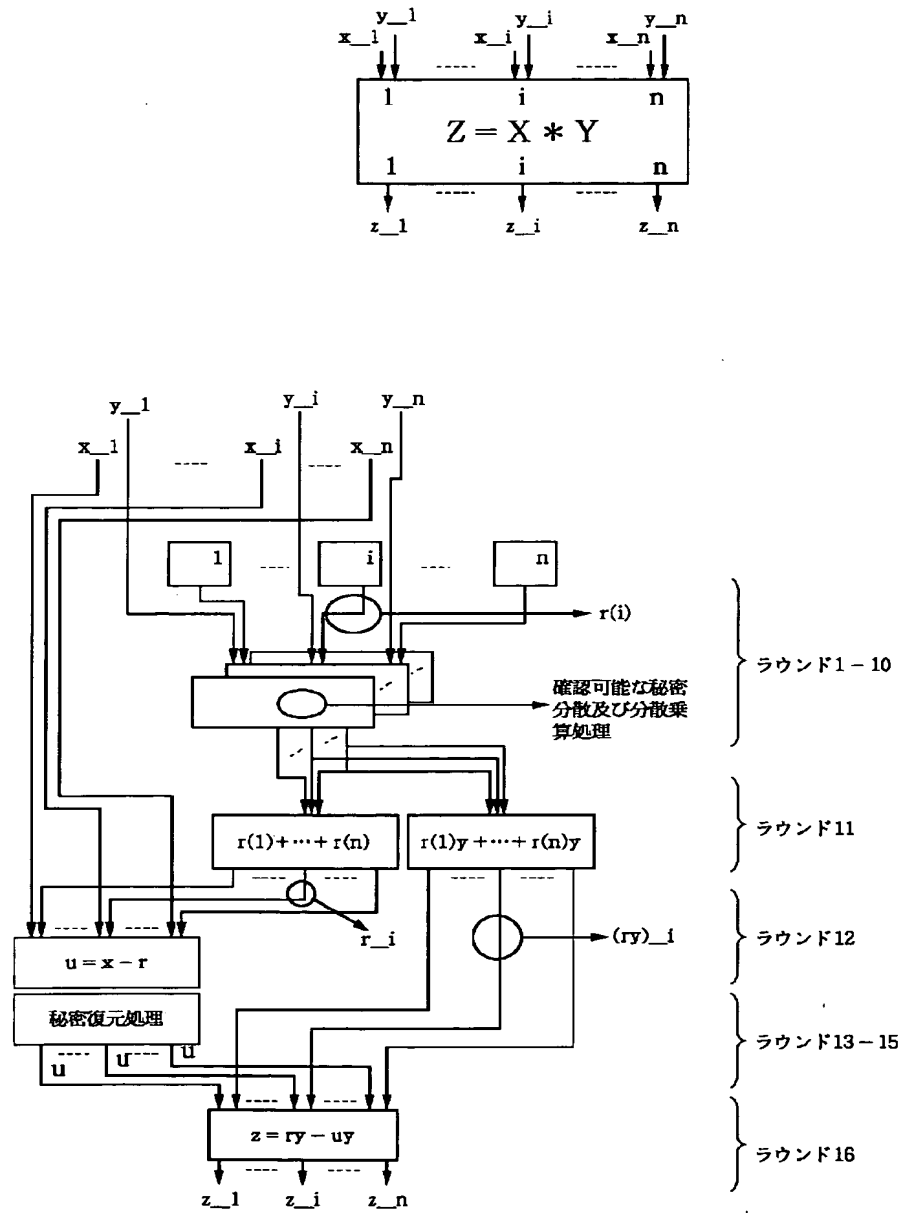
【図16】



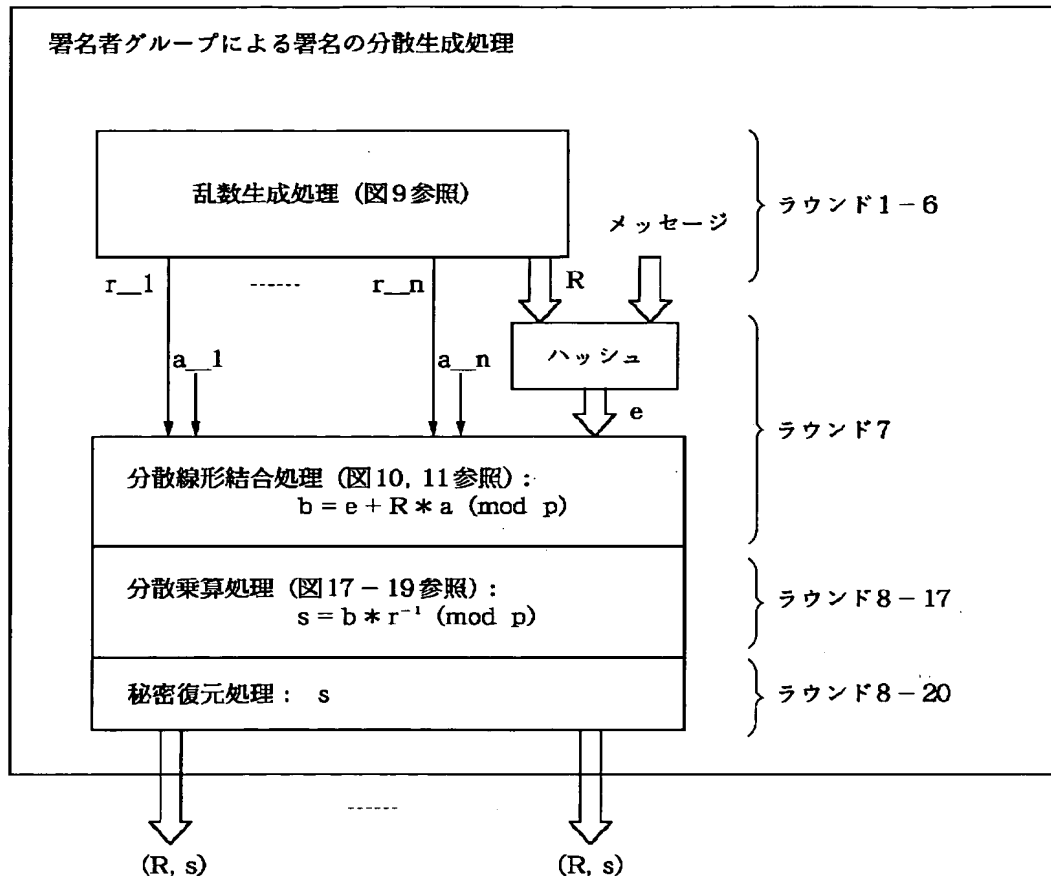
【図18】



【図19】



【図20】





## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-204697

(43)Date of publication of application : 09.08.1996

(51)Int.Cl. H04L 9/00  
H04L 9/10  
H04L 9/12  
G09C 1/00

(21)Application number : 07-008185

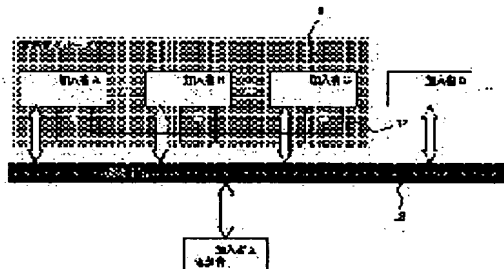
(71)Applicant : CANON INC

(22)Date of filing : 23.01.1995

(72)Inventor : MANUERU SERESED  
IWAMURA KEIICHI**(54) SIGNATURE GENERATING METHOD IN COMMUNICATION SYSTEM HAVING PLURAL EQUIPMENTS****(57)Abstract:**

**PURPOSE:** To generate a signature distributively with practical calculation amount and communication amount.

**CONSTITUTION:** In the communication system in which plural equipments are connected by a secret communication channel 13 used for information communication with secrecy to other equipments and a broadcast channel 12 sending information in common from each equipment to all other equipments, each equipment in a signature group selects 1st secrecy information at random and distributes the information with secrecy to each equipment in the group and each equipment applies 1st prescribed function on the 1st secrecy information. The obtained output is broadcast to all the equipments in the group, the 1st secrecy information is added and output values of each equipment are multiplied. Then the result of multiplication and the message are processed by using prescribed 2nd function and 2nd secrecy information is distributed by using the processing result, the result of sum and a public key to decode the distributed 2nd secret information and the decoded information is outputted as a signature together with the distribution multiplication result.

**LEGAL STATUS**

[Date of request for examination] 23.01.2002

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision  
of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office